



## Open Archive Toulouse Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of some Toulouse researchers and makes it freely available over the web where possible.

This is an author's version published in: <https://oatao.univ-toulouse.fr/18305>

**Official URL :** <http://dx.doi.org/10.1007/s00466-017-1450-z>

### To cite this version :

Mercier, Sylvain and Gratton, Serge and Tardieu, Nicolas and Vasseur, Xavier A new preconditioner update strategy for the solution of sequences of linear systems in structural mechanics: application to saddle point problems in elasticity. (2017) Computational Mechanics. pp. 1-14. ISSN 0178-7675

Any correspondence concerning this service should be sent to the repository administrator:

[tech-oatao@listes-diff.inp-toulouse.fr](mailto:tech-oatao@listes-diff.inp-toulouse.fr)

# A new preconditioner update strategy for the solution of sequences of linear systems in structural mechanics: application to saddle point problems in elasticity

Sylvain Mercier<sup>1,2</sup> · Serge Gratton<sup>3</sup> · Nicolas Tardieu<sup>1</sup> · Xavier Vasseur<sup>4</sup>

**Abstract** Many applications in structural mechanics require the numerical solution of sequences of linear systems typically issued from a finite element discretization of the governing equations on fine meshes. The method of Lagrange multipliers is often used to take into account mechanical constraints. The resulting matrices then exhibit a saddle point structure and the iterative solution of such preconditioned linear systems is considered as challenging. A popular strategy is then to combine preconditioning and deflation to yield an efficient method. We propose an alternative that is applicable to the general case and not only to matrices with a saddle point structure. In this approach, we consider to update an existing algebraic or application-based preconditioner, using specific available information exploiting the knowledge of an approximate invariant subspace or of matrix-vector products. The resulting preconditioner has the form of a limited memory quasi-Newton matrix and requires a small number of linearly independent vectors. Numerical experiments performed on three large-scale applications in elasticity highlight the relevance of the new approach. We show that the proposed method outperforms the deflation method when considering sequences of linear systems with varying matrices.

**Keywords** Deflation · Nonsymmetric matrices · Preconditioning · Saddle point matrices · Sequence of linear systems · Structural mechanics

## 1 Introduction

Finite element computations are well established strategies for the numerical simulation of material behaviour in structural mechanics. Nowadays, advanced modelling strategies are often considered to target realistic numerical simulations related to the design or safety analysis of three-dimensional geometries with heterogeneous material properties. Hence, relying on robust and efficient numerical methods for the solution of the resulting linear algebraic systems is of key importance. In this context, we consider the solution of sequences of linear systems of the form

$$A_i x_i = b_i, \quad 1 \leq i \leq \ell, \quad (1)$$

where  $A_i \in \mathbb{R}^{N \times N}$  are general nonsingular sparse matrices,  $x_i \in \mathbb{R}^N$  and  $b_i \in \mathbb{R}^N$ . These sequences appear in many applications in elasticity, e.g., when plasticity and/or viscosity effects are taken into account in the modelling. The resulting system of nonlinear equations is typically solved by a Newton-type or a Broyden-type method [34,45], leading to a sequence of the form (1). Sparse direct methods based on Gaussian elimination are usually employed for small to medium-scale problems. When the coefficient matrix is fixed, these methods are especially relevant since the factorization can be performed once for all and reused all along the sequence. When the matrices in the sequence are changing with every load or time step, preconditioned Krylov subspace methods are the methods of choice for large-scale problems [9,55,68]. Indeed, the operators in

---

✉ Xavier Vasseur  
xavier.vasseur@isae.fr

<sup>1</sup> IMSIA, UMR 9219 EDF-CNRS-CEA-ENSTA, Université Paris Saclay, 828 Boulevard des Maréchaux, 91762 Palaiseau Cedex, France

<sup>2</sup> CERFACS, 42 Avenue Gaspard Coriolis, 31057 Toulouse Cedex 1, France

<sup>3</sup> INPT-IRIT, University of Toulouse and ENSEEIHT, 2 Rue Camichel, BP 7122, 31071 Toulouse Cedex 7, France

<sup>4</sup> ISAE-SUPAERO, 10 Avenue Edouard Belin, BP 54032, 31055 Toulouse Cedex 4, France

subsequent linear systems have most often similar spectral properties. Hence, a first possible approach to design efficient numerical methods is to extract information generated during the solution of a given linear system to improve the convergence rate of the Krylov subspace method for the subsequent solutions. Deflated and augmented Krylov subspaces [14, 15, 44, 57] or Krylov subspace methods with recycling [26, 36, 50, 53, 54, 61, 67] have been proposed in this setting. Applications in structural mechanics have been provided in, e.g., [32, 33] in the symmetric positive definite case. We refer the reader to [15, 21, 22, 30, 60] for a comprehensive theoretical overview on these methods and to references therein for a summary of applications, where the relevance of these methods has been shown. An alternative consists in exploiting information generated during the solution of a given linear system to improve a given preconditioner when solving the next linear systems in the sequence. This is the main subject that we want to address in this manuscript.

Preconditioner update strategies used when solving sequences of linear systems are usually based on modifications by matrices of small rank. The main idea behind is to derive efficient preconditioners in a cheap way, thus avoiding the expensive computation of a new preconditioner in the sequence. Updates of incomplete LU based preconditioners have been considered in [7, 8, 16, 17]. Another important class of algebraically-motivated update strategies is based on improving the preconditioner by adaptive spectral information obtained directly from the Krylov subspace method, see, e.g., [4, 13, 24, 35]. Although effective, we note that both update strategies are tailored either to a specific preconditioner or to a specific Krylov subspace method.

The literature on preconditioning is extremely rich; see, e.g., the survey papers [9, 68] and references therein. In particular, many suggestions have been made concerning either purely algebraic or application based preconditioners able to exploit the structure of the matrix. Our approach markedly differs from purely preconditioning, in the sense that we propose a strategy to *update* a given preconditioner exploiting existing information. This existing information can be for instance the knowledge of matrix-vector products or of an approximate invariant subspace. This preconditioner update strategy is later referred to as limited memory preconditioner. In the numerical optimization literature, update strategies have been provided in the context of the solution of non-linear equations with quasi-Newton based methods [34, 45]. When the coefficient matrices in the sequence are symmetric positive definite, Morales and Nocedal [41] have proposed a preconditioner to be used in combination with the conjugate gradient method, which has the form of a limited memory quasi-Newton matrix; see, e.g., [42, 47] for earlier attempts. Gratton, Sartenaer and Tshimanga [28] have similarly defined a class of limited memory preconditioners based on limited memory quasi-Newton formulas that ensures good

spectral properties of the preconditioned matrix. These preconditioners require a small number  $k$  of linearly independent vectors. This family can be seen as a block variant of the Broyden–Fletcher–Goldfarb–Shanno (BFGS) updating formula for quadratic problems [45, 58]. More recently, an extension of this class of limited memory preconditioners for the solution of sequences of linear systems with symmetric indefinite matrices has been provided in [27]. Numerical experiments in structural mechanics have highlighted the relevance of the proposed limited memory preconditioner that leads to a significant decrease in terms of computational operations. Nevertheless, to the best of our knowledge, we are not aware of any formulation related to limited memory preconditioners suited to the case of a sequence of nonsymmetric linear systems. This formulation is highly relevant to improve the convergence of Krylov subspace methods, when handling sequences of linear systems, where either the original matrix or the preconditioned operator is nonsymmetric. In addition, we note that this would allow a broader class of preconditioners to be considered when handling sequence of linear systems with symmetric matrices, since the limited memory preconditioner could be then applied to the (left- or right-only) preconditioned linear system. Thus our main objectives in the manuscript are threefold: to propose a class of limited memory preconditioners adapted to the general case (1), to provide basic theoretical properties and finally to assess its performance on realistic engineering applications in structural mechanics. A detailed comparison with the deflation strategy will be also provided to emphasize the advantages of the new formulation. We emphasize that our main aim is to show the relevance of the limited memory preconditioner update strategy in combination with certain application-based preconditioners to tackle realistic applications in structural mechanics. Hence a detailed comparative performance study including advanced preconditioners of multilevel type is out of the scope of this manuscript.

The manuscript is organized as follows. In Sect. 2, we introduce the context of this study related to structural mechanics. Here, sequences of linear systems with matrices of saddle point structure have to be solved. We briefly describe the current solution method based on preconditioned Krylov subspace methods. In Sect. 3, we recall the strategy based on deflation to further accelerate the convergence of Krylov subspace methods. Then, in Sect. 4, we introduce the limited memory preconditioner, i.e., the new preconditioner update strategy. In Sect. 5, we discuss specific computational aspects and the implementation of both strategies into an existing finite element software package. Furthermore, in Sect. 6, we present numerical experiments related to three realistic applications in structural mechanics as case studies for the comparison of the different numerical methods. Concluding remarks and perspectives are finally proposed in Sect. 7.

## 2 Problem definition

In the context of structural mechanics, we briefly introduce the sequence of saddle point linear systems to be solved and describe the current numerical method that is used in practice. In the following,  $I_k \in \mathbb{R}^{k \times k}$  is the identity matrix of dimension  $k$  and  $0_{i,j} \in \mathbb{R}^{i \times j}$  refers to the zero rectangular matrix with  $i$  rows and  $j$  columns.

### 2.1 Sequence of saddle point systems

In many applications in structural mechanics, Lagrange multipliers are used to take into account specific modelling features such as the incompressibility constraint in the case of incompressible elasticity, multi-point constraints in the case of linear relationships between degrees of freedom or contact conditions between non-penetrating bodies, to name a few [11, 43]. Then, the linear system to be solved corresponds to a saddle point system of the form

$$\mathcal{K}y = c \iff \begin{pmatrix} G & B^T \\ B & 0_{m,m} \end{pmatrix} \begin{pmatrix} u \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \quad (2)$$

where  $G \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{m \times n}$ ,  $u \in \mathbb{R}^n$ ,  $\lambda \in \mathbb{R}^m$ ,  $f \in \mathbb{R}^n$  and  $g \in \mathbb{R}^m$ ,  $n$  and  $m$  being the dimensions of the discretized subspaces for the displacement and the constraints, respectively. In the case of multi-point constraints, the sparse and symmetric positive semidefinite matrix  $G$  corresponds to the stiffness matrix under the small displacement hypothesis, the essential boundary conditions being not taken into account. The dimension of the nullspace of  $G$  is then equal to the number of rigid body motions [43]. Finally, the matrix  $B$  collects the essential boundary conditions and all other constraints. Here, we also consider nonlinear mechanical problems leading to sequences of the form (with  $i = 1, \dots, \ell$ )

$$\mathcal{K}_i y_i = c_i \iff \begin{pmatrix} G_i & B^T \\ B & 0_{m,m} \end{pmatrix} \begin{pmatrix} u_i \\ \lambda_i \end{pmatrix} = \begin{pmatrix} f_i \\ g_i \end{pmatrix}, \quad (3)$$

where  $G_i \in \mathbb{R}^{n \times n}$  may be symmetric or not depending on the modelling. For instance, the nonsymmetric case occurs when a large deformation formulation is used. We further assume that  $H_i = 1/2(G_i + G_i^T)$  the symmetric part of  $G_i$  is positive semidefinite,  $B$  is of full row rank ( $\text{rank}(B) = m$ ) and that  $\mathcal{N}(H_i) \cap \mathcal{N}(B) = \{0\}$ ,  $\forall i \in \{1, \dots, \ell\}$  (where  $\mathcal{N}(B)$  denotes the nullspace of  $B$ ). These assumptions make sure the existence and uniqueness of the solution of each linear system in the sequence [10, Theorem 3.4]. To obtain an approximate solution of (3), we rely on Krylov subspace methods [37, 55, 65]. In general, Krylov subspace methods are only feasible in combination with a preconditioner when considering large-scale problems [9, 55, 68]. We cover this aspect in the next section.

### 2.2 Preconditioners for saddle point systems

We describe the two preconditioners that have been considered in the manuscript to handle the solution of saddle point systems. Due to their robustness and efficiency, both preconditioners have been retained in the application software considered in Sect. 6. We note that many preconditioners for saddle point matrices (such as augmented Lagrangian, block diagonal, block triangular, constraint and splitting-based to name a few) have been proposed in the literature. We refer the reader to, e.g., [10, 31, 46, 51] and to the monographs [19], [48, Sec. 5.2], [64, Chap. 9], [66, Chap. 9] for an extensive treatment of related preconditioning strategies.

#### 2.2.1 A block upper triangular preconditioner

When the tangent stiffness matrices  $G_i$  are symmetric positive semidefinite, we consider the block upper triangular preconditioner based on the augmented Lagrangian method (see [25] and [48, Sections 5.2.11 and 5.2.12])

$$\mathcal{M}_{t,ex} = \begin{pmatrix} G_1 + \gamma B^T B & 2B^T \\ 0_{m,n} & -\frac{1}{\gamma} I_m \end{pmatrix}, \quad \gamma > 0. \quad (4)$$

$\mathcal{M}_{t,ex}$  is related to the first matrix  $\mathcal{K}_1$  in (3) and is fixed all along the sequence. Since inverting exactly the symmetric positive definite matrix  $G_1 + \gamma B^T B$  is too demanding in terms of both computational operations and memory requirements for large-scale problems, we consider a factorized approximate preconditioner of the form  $\mathcal{M}_t = \mathcal{L}\mathcal{L}^T$  based on the incomplete Cholesky factorization of  $G_1 + \gamma B^T B$  written as  $G_1 + \gamma B^T B \approx LL^T$  [55].  $\mathcal{M}_t$  is used as a left preconditioner<sup>1</sup> leading to the following sequence of preconditioned linear systems (with  $i = 1, \dots, \ell$ )

$$A_i x_i = b_i \iff \mathcal{M}_t^{-1} \mathcal{K}_i y_i = \mathcal{M}_t^{-1} c_i. \quad (5)$$

In practice, the inverse of the preconditioner is applied using the relation

$$\mathcal{M}_t^{-1} = \begin{pmatrix} L^{-T} L^{-1} & 0 \\ 0_{m,n} & -\gamma I_m \end{pmatrix} \begin{pmatrix} I_n & 2\gamma B^T \\ 0_{m,n} & I_m \end{pmatrix}.$$

#### 2.2.2 A sparse Gaussian factorization based preconditioner

Next, we describe the default preconditioner in *Code\_Aster* [63], which consists of computing a Gaussian factorization of the saddle point matrix  $\mathcal{K}_1$  in single-precision arithmetic using the MUMPS sparse direct solver [1–3]. This factorized

<sup>1</sup> This is to be in agreement with all preconditioning strategies implemented in the host finite element software *Code\_Aster*.

approximation of  $\mathcal{K}_1$ , denoted as  $\mathcal{M}_{sp}$ , is used as a left preconditioner leading to the sequence

$$A_i x_i = b_i \iff \mathcal{M}_{sp}^{-1} K_i y_i = \mathcal{M}_{sp}^{-1} c_i, \quad (6)$$

with  $i = 1, \dots, \ell$ . The application of  $\mathcal{M}_{sp}^{-1}$  is performed through successive backward and forward substitutions and, unless stated otherwise,  $\mathcal{M}_{sp}$  is kept fixed all along the sequence (6).

Whatever the preconditioner ( $\mathcal{M}_t$  or  $\mathcal{M}_{sp}$ ), we do obtain a sequence of preconditioned linear systems of the form (1); see either Eqs. (5) or (6). In the following, we select the restarted GMRES method [56] (later denoted by GMRES( $m$ )) as a Krylov subspace method to handle this sequence of systems involving nonsymmetric matrices. Even with the block upper triangular or the sparse Gaussian factorization based preconditioners described above, the convergence of preconditioned GMRES can be slow for specific large-scale systems. Thus, we aim at proposing efficient and robust solution strategies able to enhance the convergence rate of restarted GMRES. We first consider a popular strategy based on deflation in such a setting. Deflation will then serve as a reference advanced numerical method through the manuscript.

### 3 Strategy based on deflation

We first introduce notation used in Sects. 3 and 4, respectively. Given a matrix subspace  $\mathcal{S}$  of finite dimension, we denote the range of  $\mathcal{S}$  by  $\mathcal{R}(\mathcal{S})$ , the null space of  $\mathcal{S}$  by  $\mathcal{N}(\mathcal{S})$  and the orthogonal complement of  $\mathcal{S}$  by  $\mathcal{S}^\perp$ . If  $\mathcal{V}$  and  $\mathcal{W}$  are complementary subspaces of a vector space, we denote by  $P_{\mathcal{V}, \mathcal{W}}$  the projection operator onto  $\mathcal{V}$  along  $\mathcal{W}$ <sup>2</sup>. Finally  $P_{\mathcal{V}, \mathcal{V}^\perp}$ , the orthogonal projection operator onto  $\mathcal{V}$ , is denoted by  $P_{\mathcal{V}}$ .

#### 3.1 Case of a single linear system

We first consider a single linear system denoted by  $Ax = b$  with  $A \in \mathbb{R}^{N \times N}$ ,  $x \in \mathbb{R}^N$  and  $b \in \mathbb{R}^N$ . The deflation technique to accelerate the convergence of Krylov subspace methods for the solution of a given linear system has been known for a long time; see, e.g., [14, 15, 44] and the extensive bibliography proposed in [29]. Applications to structural mechanics have been provided in, e.g., [32, 33] in the symmetric positive definite case. In the last decade, deflation has been used and analyzed in combination with multigrid and domain decomposition methods, which results in efficient algorithms [32, 62]. Extensions to nonsymmetric or

non-Hermitian problems have been provided in, e.g., [20–22, 30]. In the following, we assume that  $S \in \mathbb{R}^{N \times k}$  is of full column rank  $k$ , with  $k \leq N$  and we denote by  $\mathcal{S} = \mathcal{R}(S)$ . We introduce the orthogonal projection operator  $P_{(A, \mathcal{S})^\perp}$

$$P_{(A, \mathcal{S})^\perp} = I_N - AS(S^T A^T AS)^{-1} S^T A^T, \quad (7)$$

and the oblique projection operator  $Q := P_{(A^T A, \mathcal{S})^\perp, \mathcal{S}}$

$$Q = I_N - S(S^T A^T AS)^{-1} S^T A^T A. \quad (8)$$

We note that  $Q$  defined in Eq. (8) satisfies the relation

$$AQ = P_{(A, \mathcal{S})^\perp} A. \quad (9)$$

In our setting, deflation consists of solving the projected problem

$$P_{(A, \mathcal{S})^\perp} A \tilde{x}_D = P_{(A, \mathcal{S})^\perp} b, \quad (10)$$

which is supposed to be much easier to solve by the restarted GMRES method than the original linear system  $Ax = b$ . Theorem 5.1 and Corollary 5.3 of [22] show that the GMRES method applied to the singular linear system (10) is well defined at each iteration step for every initial guess and terminates with a solution of the system if and only if  $\mathcal{S} \cap (A, \mathcal{S})^\perp = \{0\}$ . We refer the reader to [22, 30] for a theoretical analysis of the convergence of the deflated GMRES method. We note that the solution of the original linear system  $Ax = b$  is simply recovered by

$$x_D = S(S^T A^T AS)^{-1} S^T A^T b + Q \tilde{x}_D.$$

Indeed, a direct calculation gives

$$b - Ax_D = P_{(A, \mathcal{S})^\perp} b - P_{(A, \mathcal{S})^\perp} A \tilde{x}_D,$$

i.e., the true residual equals to the residual of the deflated linear system (10).

#### 3.2 Case of a sequence of linear systems

With the same matrix  $S$  as in Sect. 3.1, we consider the strategy based on deflation to solve the sequence of linear systems (5) or (6). In this setting, we note that the projection operators involved in deflation need to be defined *for each linear system* of the sequence. Hence, we introduce the orthogonal projection operators

$$P_{(A_i, \mathcal{S})^\perp} = I_N - A_i S(S^T A_i^T A_i S)^{-1} S^T A_i^T, \quad (11)$$

and the oblique projection operators  $Q_i := P_{(A_i^T A_i, \mathcal{S})^\perp, \mathcal{S}}$

$$Q_i = I_N - S(S^T A_i^T A_i S)^{-1} S^T A_i^T A_i. \quad (12)$$

<sup>2</sup>  $P_{\mathcal{V}, \mathcal{W}}$  is the unique projection operator with range  $\mathcal{R}(P_{\mathcal{V}, \mathcal{W}}) = \mathcal{V}$  and null space  $\mathcal{N}(P_{\mathcal{V}, \mathcal{W}}) = \mathcal{W}$  [39].



Deflated GMRES( $\underline{m}$ ) is then used through the whole sequence as

$$P_{(A_i S)^\perp} A_i \tilde{x}_i = P_{(A_i S)^\perp} b_i, \quad (13)$$

the solution being given by

$$x_i = S(S^T A_i^T A_i S)^{-1} S^T A_i^T b_i + Q_i \tilde{x}_i, \quad (14)$$

with  $2 \leq i \leq \ell$ . We assume that the conditions  $\mathcal{S} \cap (A_i \mathcal{S})^\perp = \{0\}$  hold. This guarantees a breakdown-free GMRES( $\underline{m}$ ) computation when solving the sequence of deflated linear systems.

## 4 Strategy based on preconditioner update

We introduce, in this section, the new strategy based on preconditioner update to solve the given sequence of linear systems (5) or (6).

### 4.1 Case of a single linear system

In this setting, we first define the class of limited memory preconditioners (LMP).

**Definition 1** Let  $A \in \mathbb{R}^{N \times N}$  be a nonsingular matrix and assume that  $S \in \mathbb{R}^{N \times k}$  is of full column rank  $k$ , with  $k \leq N$ . The matrix  $H \in \mathbb{R}^{N \times N}$  defined as

$$H = (I_N - AS(S^T A^T AS)^{-1} S^T A^T) + S(S^T A^T AS)^{-1} S^T A^T \quad (15)$$

is called the limited memory preconditioner.

When  $A$  corresponds to a preconditioned operator [as in (5) or (6)],  $H$  defines a second level preconditioner aiming to further improve the convergence rate of the Krylov subspace method. The motivation behind this definition can be briefly explained. In numerical optimization, the application of certain quasi-Newton methods [34, 45] to the numerical solution of  $F(x) = 0$  with  $F(x) = Ax - b$  provides an approximation of the inverse of the Jacobian (here  $A^{-1}$ ). Hence, we aim at using this approximation as a candidate preconditioner for Krylov subspace methods when solving  $Ax = b$ . This idea has been notably exploited in [69, 70], where the construction of the preconditioner was based on rank-1 updates issued from either the Broyden's method [12] or the Eirola and Nevanlinna's method [18]. The relation (15) in Definition 1 corresponds to a block rank- $k$  extension of one of these updates and has been derived in [38, Proposition 3.1.1]; see also [38, Section 3.1] for further details. In the following, our main purpose is to study and investigate the potential of the matrix  $H$  when considered as a preconditioner of  $A$ .

In the framework of Krylov subspace methods, it is usually important to first characterize the properties of the preconditioner [55]. In our setting, the preconditioner  $H$  given in Definition 1 is provably nonsingular if and only if

$$\mathbb{R}^N = (A\mathcal{S})^\perp \oplus \mathcal{S}. \quad (16)$$

This is stated in Theorem 1 given in ‘‘Appendix’’. Later, we will thus promote the selection of  $S \in \mathbb{R}^{N \times k}$  such that the relation (16) holds. For example, this occurs when considering exact spectral information of  $A$ , i.e., when  $\mathcal{S} = \mathcal{R}(S)$  spans the eigenspace of  $A$  associated with  $\{\lambda_1, \dots, \lambda_k\}$  which contains some real eigenvalues and possibly some pairs of complex eigenvalues and their conjugate<sup>3</sup>. This leads to the so called spectral LMP variant.

The next step is to analyze the spectral properties of the preconditioned operator  $AH$ . In the following, we denote by  $\Lambda(M)$  the set of eigenvalues of a given square matrix  $M$ ,  $M \in \mathbb{R}^{N \times N}$ . Furthermore, we assume that the columns of  $W \in \mathbb{R}^{N \times k}$  and  $W_\perp \in \mathbb{R}^{N \times (N-k)}$  form an orthonormal basis for  $A\mathcal{S}$  and  $(A\mathcal{S})^\perp$ , respectively. Our central result can then be stated as follows: the spectrum of the preconditioned operator  $AH$  is given by

$$\Lambda(AH) = \{1\} \cup \Lambda(W_\perp^T A W_\perp). \quad (17)$$

This is shown in Theorem 2 given in ‘‘Appendix’’. We deduce from Theorem 2 that 1 is an eigenvalue of  $AH$  of multiplicity at least  $k$ , whereas the other eigenvalues belong to the spectrum of  $W_\perp^T A W_\perp$ . When exact eigenvectors of  $A$  are selected as columns of  $S$ , a simpler form of Theorem 2 can be deduced; see [38, Corollary 3.2.3]. However, the case of the spectral limited memory preconditioner is only of limited interest. In practice, for large-scale problems, computing exact spectral information is considered as too computationally expensive. We will thus have to rely on a different choice for the column vectors of  $S$  as also proposed in the deflation literature. We will later cover this aspect in Sect. 5.3.

### 4.2 Case of a sequence of linear systems

The same column vectors of  $S$  are used through the sequence, unless stated otherwise. Hence, in this context, the limited memory preconditioner is simply given by

<sup>3</sup> We indeed use the property  $\text{span}\{s, \bar{s}\} = \text{span}\{\text{Re}(s), \text{Im}(s)\}$  to justify the fact that the entries of  $S$  can be chosen as real-valued, as stated in Definition 1; see Sect. 5.3 for a detailed discussion.

$$H = I_N - A_1 S (S^T A_1^T A_1 S)^{-1} S^T A_1^T + S (S^T A_1^T A_1 S)^{-1} S^T A_1^T. \quad (18)$$

GMRES( $\underline{m}$ ) combined with this *fixed* right preconditioner is then used through the whole sequence as

$$A_i H \tilde{x}_i = b_i, \quad x_i = H \tilde{x}_i, \quad 2 \leq i \leq \ell. \quad (19)$$

We point out here the main difference with the strategy based on deflation: a fixed preconditioner is used through the sequence, whereas the update of projection operators is mandatory in the deflation approach (see Sect. 3.2).

## 5 Computational cost, memory requirements, selection of $S$ and implementation details

We now examine the computational cost and memory requirements of the two strategies proposed in Sects. 3 and 4, respectively. We conclude this section by giving details on the choice of  $S$  required in both approaches and on the implementation.

### 5.1 Strategy based on preconditioning

In the following, we propose a possible implementation of the limited memory preconditioner. Due to [38, Proposition 3.1.2], we know that  $H$  is invariant by right multiplication of  $S$  with a nonsingular matrix. Hence, our approach consists of replacing  $S$  with  $Z$  such that  $\mathcal{R}(Z) = \mathcal{R}(S)$  and  $Z^T A^T A Z = I_N$ . To do so, we apply a Gram-Schmidt procedure to produce a  $A^T A$ -conjugate basis of  $\mathcal{R}(S)$ . With  $X = AZ$  and  $Y = Z - X$ ,  $H$  can then be simply written as

$$H = I_N + Y X^T. \quad (20)$$

Algorithm 1 summarizes the construction of the limited memory preconditioner, with  $Z = [z_1, \dots, z_k]$  and  $X = [x_1, \dots, x_k]$ , respectively. This algorithm requires, in terms of computational cost,  $k$  matrix vector products with  $A$  and  $4N + \sum_{i=1}^{k-1} (6iN + 6N) + kN = (3k^2 + 4k - 2)N$  additional floating point operations. In terms of memory requirements,  $2k$  vectors of length  $N$  are required to store both  $X$  and  $Z$ . Each application of  $H$  on a vector of appropriate dimension can be directly performed in  $(4k + 1)N$  floating point operations as shown in Eq. (20). Finally, we note that the construction of the limited memory preconditioner can be handled in a matrix-free environment, i.e., only matrix vector products with  $A$  are required as shown in Algorithm 1. We denote by  $\|\cdot\|_2$  the Euclidean norm and by  $S = [s_1, \dots, s_k]$  the columns of  $S$  in Algorithm 1.

---

**Algorithm 1** Computation of  $Y, X \in \mathbb{R}^{N \times k}$  such that  $H = I_N + Y X^T$ .

---

```

1:  $z_1 = s_1$ 
2:  $x_1 = A z_1$ 
3:  $\sigma = \|x_1\|_2$ 
4:  $x_1 = x_1 / \sigma$ 
5:  $z_1 = z_1 / \sigma$ 
6: for  $i = 1$  to  $k - 1$  do
7:    $z_{i+1} = s_{i+1}$ 
8:    $x_{i+1} = A z_{i+1}$ 
9:    $f = X_i^T x_{i+1}$ 
10:   $z_i = z_{i+1} - Z_i f$ 
11:   $x_i = x_{i+1} - X_i f$ 
12:   $\sigma = \|x_{i+1}\|_2$ 
13:   $x_{i+1} = x_{i+1} / \sigma$ 
14:   $z_{i+1} = z_{i+1} / \sigma$ 
15: end for
16:  $Y = X - Z$ 
```

---

### 5.2 Strategy based on deflation

The implementation of the deflation method that we have adopted is also based on the Gram-Schmidt procedure; see [38, Section 3.4.3] for additional details. Table 1 summarizes the main computational costs involved in both strategies. This excludes the common cost related to the computation of the column vectors of  $S$ . As pointed out in Sect. 4.2, the main difference lies in the construction phase, since the limited memory preconditioner is built only once.

### 5.3 Choice of $S$

In Sect. 4.2, the spectrum of the preconditioned operator  $AH$  has been determined. As discussed above, in the context of large-scale applications, computing exact spectral information is usually considered as too expensive. It is thus common to rely on approximate spectral information based on Ritz vectors [49] as often proposed in the literature related to eigenvalue computations [5]. For the sake of completeness, we briefly define the notion of Ritz pairs.

**Definition 2** Let  $A$  be a nonsingular matrix and assume that  $l$  iterations of GMRES( $\underline{m}$ ) have been performed so that the relation  $V_l^T A V_l = H_l$  holds with  $V_l \in \mathbb{R}^{N \times l}$  and  $H_l \in \mathbb{R}^{l \times l}$  ( $l \leq \underline{m}$ ). A Ritz pair is defined as a pair  $(w = V_l y, \theta) \in \mathbb{C}^N \times \mathbb{C}$  where  $(y, \theta) \in \mathbb{C}^l \times \mathbb{C}$  is an eigenpair of  $H_l$ .  $w$  is called the Ritz vector associated to the Ritz value  $\theta$ .

In the case of a sequence of linear systems, we extract the Ritz information during the last complete cycle of GMRES( $\underline{m}$ ) when solving the first linear system  $A_1 x_1 = b_1$ . We then select  $k$  Ritz vectors associated with the  $k$  smallest in modulus Ritz values (with  $k \leq \underline{m}$ ). The same column vectors of  $S$  are used through the sequence. We note that these Ritz vectors can be complex-valued since  $H_l$  is nonsymmetric. In case of a Ritz pair  $(y, \theta)$  with a complex-valued Ritz value

**Table 1** Computational cost related to the construction and application phases of the limited memory preconditioner and storage requirements

	Construction	Application	Storage
LMP	$(3k^2 + 4k - 2)N + k C_A$	$(4k + 1)N$	$2kN$
Deflation	$(\ell - 1)(3k^2 + 3k - 2)N + (\ell - 1)k C_A$	$(4k + 1)N$	$2kN$

A similar information is provided for the deflation method. The computational cost of a matrix-vector product with  $A$  is denoted by  $C_A$

$\theta$ ,  $H_l$  being real-valued, we observe that  $(\bar{y}, \bar{\theta})$  is also a Ritz pair. Since the real and imaginary parts of the Ritz vector  $y$  generate the same subspace as the two conjugate Ritz vectors (i.e.  $\text{span}\{Re(y), Im(y)\} = \text{span}\{y, \bar{y}\}$ ), we impose to select both real and imaginary parts of the Ritz vector  $y$  as columns of  $S$ . This allows us to keep real arithmetic in the definition of the LMP preconditioner (see Definition 1). Furthermore, depending on the value of  $k$ , we may in fact select  $k + 1$  columns in  $S$  to include both  $Re(y)$ ,  $Im(y)$  if necessary. Finally, in case of any linearly dependent column(s) in  $S^4$ , we simply remove such column(s). This makes sure that  $S$  is of full column rank as stated in Definition 1. We refer the reader to [38, Section 3.4.1] for additional details. Since  $S^T A S = \Theta$ , with  $\Theta \in \mathbb{C}^{k \times k}$  being the diagonal matrix with the  $k$  Ritz values as entries, we note that  $H$  is nonsingular since relation (16) is satisfied.

## 5.4 Implementation details

The different strategies based on deflation and on preconditioning have been implemented in the framework of the open-source software *Code\_Aster*<sup>5</sup> (version 12.3.0), which is a general purpose finite element code developed at EDF (the French utility, Electricité de France) through the PETSc<sup>6</sup> library (version 3.4.5). We refer the reader to [38, Appendix A] for a complete description of the code and of the corresponding data structure.

## 6 Numerical experiments

The purpose of this section is to illustrate the efficiency of the limited memory preconditioner on various sequences of linear systems with matrices of saddle point structure in structural mechanics. A large-scale problem with a fixed matrix and multiple right-hand sides is first considered in Sect. 6.1, while two sequences of linear systems will be analyzed later in Sects. 6.2 and 6.3, respectively.

We use GMRES(30) to solve the sequence of linear systems. Selecting this large value of the restart parameter in

GMRES( $m$ ) notably allows us to investigate the influence of the number of Ritz vectors on the performance of the limited memory preconditioner. We choose the following stopping criterion

$$\frac{\|b_i - A_i x_i\|_2}{\|b_i\|_2} \leq 10^{-8}, \quad 1 \leq i \leq \ell, \quad (21)$$

with a zero initial guess  $x_i^0$ . Application of GMRES(30) for the solution of either (5) or (6) is later referred to as “Standard”. As explained in Section 2,  $A_i$  correspond to preconditioned operators. This reference method will be compared with the advanced “Deflation” and the “Ritz-LMP” strategies, described in Sects. 3 and 4, respectively.

The numerical results have been obtained on Aster5, a IBM IDATAPLEX computer located at EDF R&D Data Center (each node of Aster5 is equipped with 2 Intel Xeon E5 – 2600, each running 12 cores at 2.7 Ghz). Physical memory available on a given node (24 cores) of Aster5 ranges from 64 GB to 1 TB. The code has been compiled by the Intel compiler suite with the best optimization options and linked with the Intel MKL BLAS and LAPACK subroutines. We report both iteration counts and measure of computational effort in seconds. We note that this numerical study has been performed in a serial environment and we refer the reader to [38, Section 3.4.5] for additional numerical experiments on a parallel distributed memory computer, where the efficiency of the proposed strategy has been shown as well.

### 6.1 Containment building of a nuclear reactor

In this section, we investigate the mechanical properties of the containment building of a nuclear reactor of a Pressurized Water Reactor power plant (see Fig. 1). This building protects both the reactor from external aggressions and the environment if an internal accident occurs. Robust and accurate numerical simulations are thus required for both design and safety analysis. We consider an advanced mechanical modelling that takes into account numerous prestressing tendons, whose role is to improve the global resistance of the structure. The containment building is subject to gravity and to an internal pressure. The whole loading is gradually applied into 4 successive steps. Each pitch of loading then corresponds to a specific linear system in the sequence, where only the

<sup>4</sup> We emphasize that this situation did not occur in all our numerical experiments.

<sup>5</sup> <http://www.code-aster.org>.

<sup>6</sup> <http://www.mcs.anl.gov/petsc/>.



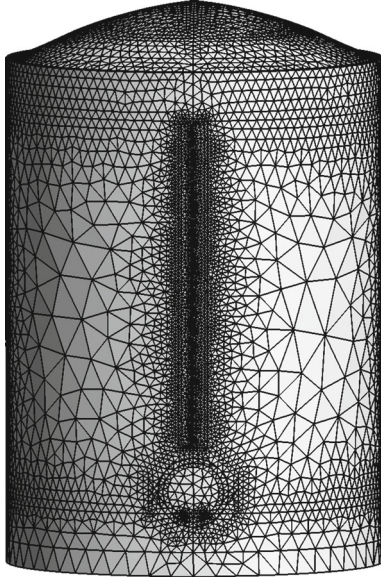


Fig. 1 Containment building: three-dimensional mesh

right-hand side has changed (i.e.  $A_1 = \dots = A_4$ ). The introduction of Lagrange multipliers stems from the imposition of kinematic relations modelling perfect adhesion between the prestressing tendons and the concrete [40] and to the dualization of the essential boundary conditions. In this setting,  $B$  admits either five or one nonzero entries per row, respectively. This study is known to be complex for different reasons. First, from a mechanical point of view, the modelling (which leads to a symmetric positive semidefinite  $G_1$  matrix) is rather advanced with a mixing of three-dimensional elements for the concrete, two-dimensional elements for the metal shell placed on the intern wall of the building (to ensure the sealing if an accidental leak occurs), and of one-dimensional elements for the tendons. Moreover, since the prestressing tendons are attached to the concrete thanks to dualized linear relations, the number of Lagrange multipliers is really large ( $m = 158,928$  for a global size of  $N = 442,725$ ). The number of nonzero entries of  $G_1$  and  $G_1 + \gamma B^T B$  is 7,079,238 and 8,343,685, respectively. Secondly, the occurrence of a large number of prestressing tendons (more than 600 here) induces a nullspace of large dimension for the stiffness matrix. Actually, it is known that this dimension is related to the number of rigid body motions of the subbodies of materials contained within the finite element mesh [43]. This numerical study is thus challenging and serves as a relevant realistic test case in solid mechanics to investigate the efficiency of preconditioners for Krylov subspace methods.

In this numerical experiment, we consider the block upper triangular preconditioner  $\mathcal{M}_l$  described in Section 2.2.1. As explained in [38, Section 1.3.2.2], we set  $\gamma$  to  $2,4684 \times 10^{11}$  and consider a level of fill-in equal to 8 in the incomplete Cholesky factorization of the (1, 1) block of  $\mathcal{M}_l$ .

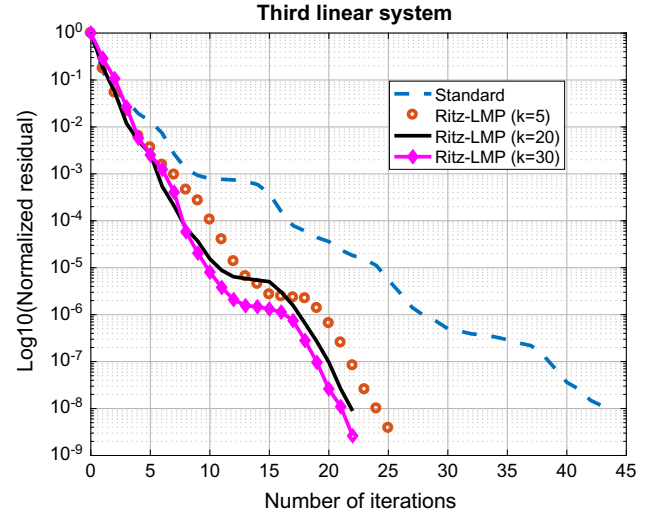


Fig. 2 Containment building: convergence history of preconditioned GMRES(30) for the third linear system in the sequence. Case of limited memory preconditioners with  $k = 5, 20$  or 30 Ritz vectors

Table 2 Containment building (sequence of three linear systems where only the right-hand sides change): cumulative iteration count, computational time and memory requirements for different limited memory preconditioners

Strategy	Standard	Ritz-LMP		
		$k = 5$	$k = 20$	$k = 30$
Total iteration count	135	74	66	64
CPU time (s)	75.6	42.9	40.2	36.9
CPU time decrease (%)	×	43	47	51
Memory (Mo)	6895	6950	7071	7172
Memory increase (%)	×	0.8	2.6	4

Case of  $k = 5, 20$  or 30 Ritz vectors. “Standard” corresponds to the solution of the linear system (5) with GMRES(30) combined with the block triangular preconditioner  $\mathcal{M}_l$

We first consider the Ritz-LMP strategy with a growing number of Ritz vectors ( $k = 5, 20, 30$ , respectively). Figure 2 shows the convergence history of GMRES(30) for the third linear system in the sequence. A very similar behaviour is found for the second and fourth linear systems. In addition, we show in Table 2 the cumulative iteration count over the three linear systems, the measure of computational effort in seconds<sup>7</sup> and the memory requirements provided by PETSc, respectively. We note that the smallest number of iterations is obtained when a large value of Ritz vectors ( $k = 30$ ) is selected. Whatever the value of  $k$ , the limited memory preconditioner is found to significantly improve the convergence of the preconditioned Krylov subspace method (Standard vs Ritz-LMP); the computational timings are divided by a factor

<sup>7</sup> The CPU time for the Ritz-LMP strategy does include the cost of forming  $X, Y$  as required in (20).

**Table 3** Containment building (sequence of three linear systems where only the right-hand sides change): cumulative iteration count, computational time and memory requirements for different deflation methods

Strategy	Standard	Deflation		
		$k = 5$	$k = 20$	$k = 30$
Total iteration count	135	74	64	59
CPU time (s)	75.6	42.6	38	33.6
CPU time decrease (%)	×	43	50	55.6
Memory (Mo)	6895	6950	7071	7172
Memory increase (%)	×	0.8	2.6	4

Case of  $k = 5, 20$  or  $30$  Ritz vectors. “Standard” corresponds to the solution of the linear system (5) with GMRES(30) combined with the block triangular preconditioner  $\mathcal{M}_t$ .

greater than 2 for  $k = 30$ , which is a very satisfactory result. Moreover, this improvement comes at a price of a quite low increase in memory requirements (at most 4%).

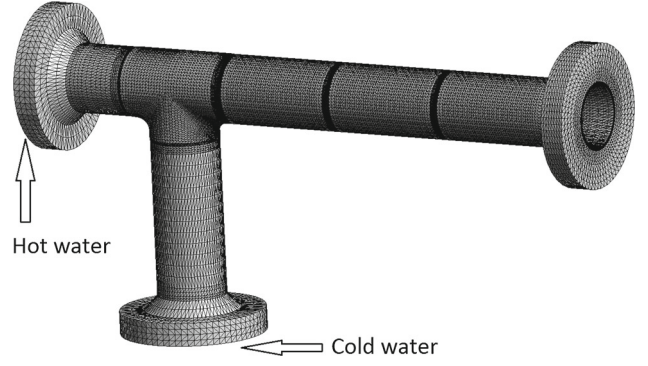
Table 3 shows the results related to the deflated GMRES(30) method using exactly the same Ritz vectors as in the previous numerical experiment. We note that the deflation method leads to slightly better reductions in terms of iterations or computational timings, whereas memory requirements are found to be similar as expected.

Finally, we note that both strategies significantly improve the results of [27]. In [27, Section 4.3], a symmetric positive definite block diagonal preconditioner and a limited memory preconditioner tailored to symmetric indefinite matrices were combined to tackle the same challenging problem. The combination of GMRES(30) with the block diagonal preconditioner required 509 iterations and 315 seconds to satisfy the same stopping criterion (see Table 2.1 of [38] for the corresponding timings). A huge acceleration factor (up to 8.5) is thus obtained in our new setting. Indeed, the new class of limited memory preconditioner offers the advantage to allow a broader class of preconditioners to be considered, as nicely illustrated here.

## 6.2 Shut down nuclear reactor cooling loop

In Sect. 6.1, we have considered the case of a sequence with a fixed matrix. Next, we investigate the performance of the Ritz-LMP strategy on sequences arising from Newton’s method with varying matrices. From now on, we use the single-precision arithmetic sparse Gaussian factorization based preconditioner  $\mathcal{M}_{sp}$  introduced in Sect. 2.2.2.

The selected problem is related to the modelling of the thermal fatigue of a shut down nuclear reactor cooling loop. The heat from the nuclear core is extracted by circulation of pressurized water in the primary circuit. When the reactor is shut down, the studied cooling loop allows to evacuate the heat of the primary circuit and the residual power of the



**Fig. 3** Mesh of the shut down nuclear reactor cooling loop

nuclear fuel, using cold waters. Some oscillations of the temperature can occur at locations where both cold and hot waters are in contact and it is necessary to model the resistance of the component to the thermal fatigue. The computational mesh is illustrated in Fig. 3.

The finite element discretization involves  $N = 571,281$  unknowns, with less than 1% of Lagrange multipliers, corresponding to the essential dualized boundary conditions. Newton’s method is employed because of the nonlinearity of the constitutive law of the structure (Chaboche elasto-viscoplastic law [52]), which leads to a sequence of the form (6) with 67 linear systems (with  $G_i$  being symmetric positive semidefinite). This problem is challenging due to the condition number of the saddle point matrix  $\mathcal{K}_1$ , which is of order of  $10^{10}$ .<sup>8</sup> Indeed, it is known that this value can be related to the properties of the mesh; in particular, when strong spatial variations in the size of the mesh elements occur, or when some of them are flattened, the condition number is found to increase [59]. The former case arises at the intersection of both pipes (see Figure 3). In general, when the condition number is moderate, GMRES(30) combined with the  $\mathcal{M}_{sp}$  preconditioner requires a quite low number of iterations to satisfy the stopping criterion (21). In the case of a large condition number, this iteration count is often found to increase, and we expect that the Ritz-LMP strategy may provide an improvement.

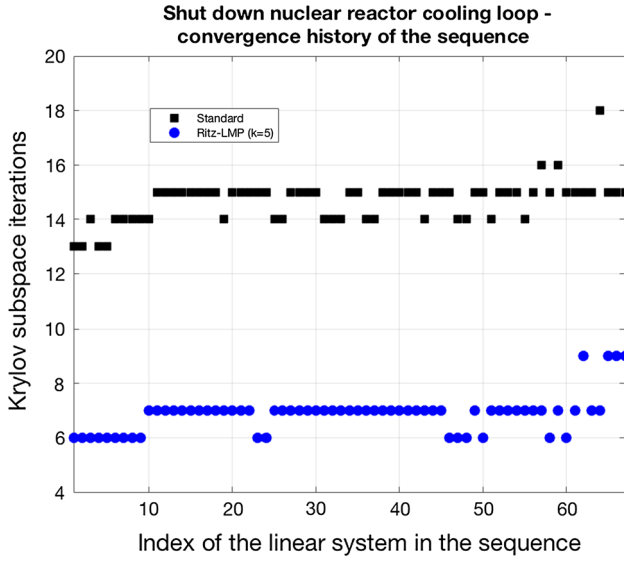
We comment on the results related to the solution of the complete sequence, including the sparse Gaussian factorization phase. Table 4 collects the results obtained with three different solution methods: the standard approach, the Ritz-LMP strategy and the deflation method with  $k = 5$  vectors. Indeed, in this case, the value of  $k = 5$  seems to be a reasonable choice, since the number of GMRES(30) iterations when combined with  $\mathcal{M}_{sp}$  usually remains relatively low (i.e. less than the restart parameter  $\underline{m}$  (30), see Fig. 4) and

<sup>8</sup> This estimate has been obtained as a by-product of the application of MUMPS (here in double-precision arithmetic) to the first matrix  $\mathcal{K}_1$  in the sequence.

**Table 4** Shut down nuclear reactor cooling loop (sequence of 67 linear systems): cumulative iteration count, computational time and memory requirements for both the Ritz-LMP and the deflation methods with  $k = 5$  Ritz vectors, respectively

Strategy	Standard	Ritz-LMP	Deflation
Total iteration count	983	460	410
CPU time (s)	961	578	873
CPU time decrease (%)	×	39.9	9.2
Memory (Mo)	14074	14117	14117
Memory increase (%)	×	0.03	0.03

“Standard” corresponds to the solution of the linear system (6) with GMRES(30) combined with the sparse Gaussian factorization based preconditioner  $\mathcal{M}_{sp}$

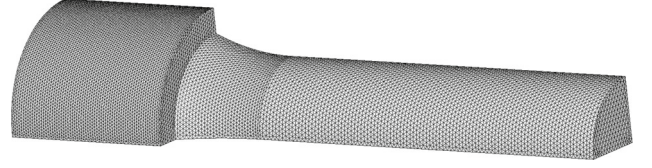


**Fig. 4** Shut down nuclear reactor cooling loop: number of Krylov subspace iterations versus index of the linear system in the sequence for both the “Standard” and Ritz-LMP approaches

a larger value of  $k$  does not imply a significant additional gain in terms of computational time. We first note that the use of either the Ritz-LMP or the deflation approach is very efficient in terms of reduction of Krylov subspace iterations. Concerning the computational effort, the Ritz-LMP strategy is much more efficient than the deflation method. This can be explained by the fact that the deflation operator  $P_{(A_i S)^\perp}$  as well as  $Q_i$  [see Eqs. (11) and (12)], need to be updated at each new linear system, using the new matrix  $A_i$ ; see Table 1. This numerical experiment clearly illustrates the interest of the limited memory preconditioning strategy in this setting.

### 6.3 Tensile test

The previous numerical experiment has highlighted the efficiency of the Ritz-LMP strategy. In particular, this strategy



**Fig. 5** Mesh of the tensile

has been shown to be relevant compared to deflation in terms of computational time, since the preconditioner is fixed through the whole sequence. While, in Sects. 6.1 and 6.2, the tangent stiffness matrices  $G_i$  were symmetric positive semidefinite, we further investigate the performance of the limited memory preconditioner, when the modelling leads to original matrices  $\mathcal{K}_i$  that are nonsymmetric.

The tensile test problem illustrates a tensing test on a cylindrical body. Due to axial and radial symmetries, the finite element modelling considers only 1/8th of the structure, as shown in Fig. 5. We employ a nonlinear constitutive law of the material (elasto-plastic law of Von Mises with isotropic hardening), written in the large deformation formulation proposed by Simo and Miehe [6]. The total number of unknowns is  $N = 977,365$  including about 10% of Lagrange multipliers corresponding to several symmetry essential boundary conditions (i.e. a zero normal displacement). Here, the preconditioner is based on a sparse Gaussian factorization of a modified matrix deduced from  $\mathcal{K}_1$ , where some entries of  $\mathcal{K}_1$  have been filtered out in order to reduce the computational time related to the factorization and the solution phases. An important feature has to be taken into account in this setting. By default, in *Code\_Aster*, the sparse Gaussian factorization based preconditioner  $\mathcal{M}_{sp}$  is used during the complete sequence. However, if the Krylov subspace method requires too many iterations (more than 100 here) to satisfy the stopping criterion (21) for a given system  $A_i x_i = b_i$ , a new sparse Gaussian factorization of the next matrix in the sequence  $\mathcal{K}_{i+1}$  is performed. In such a case,  $H$  is discarded after the solution of the  $i$ -th linear system, and a new limited memory preconditioner is defined at the end of the solution of  $A_{i+1} x_{i+1} = b_{i+1}$  with new Ritz vectors. We emphasize that this situation did not happen in the previous two numerical experiments.

The loading is incrementally applied on the structure and one nonlinear problem is solved with the Newton’s method at each loading step to find the local equilibrium state. First, we impose 14 increments, which leads to 61 cumulated Newton iterations. Table 5 collects the results of this first numerical experiment. We note that the computational time also includes the time spent in the factorization phases. Using the Ritz-LMP strategy does lead to important savings in terms of both iteration count (30%) and computational time (33%). Indeed, the use of the limited memory preconditioner keeps the number of Krylov iterations below 100 iterations most



**Table 5** Tensile test with 14 loading increments: cumulative Newton and Krylov iteration counts, number of sparse Gaussian factorizations, computational time and memory requirements for the limited memory preconditioner with  $k = 5$  Ritz vectors

Strategy	Standard	Ritz-LMP
Loading steps	14	14
Newton iteration count	61	61
Total Krylov iteration count	3218	2259
Number of sparse Gaussian factorizations	3	2
CPU time	5535	3717
CPU time decrease (%)	×	33
Memory (Mo)	14982	15011
Memory increase (%)	×	0.2

“Standard” corresponds to the solution of the linear system (6) with GMRES(30) combined with the sparse Gaussian factorization based preconditioner  $\mathcal{M}_{sp}$

**Table 6** Tensile test with at least 5 loading increments: cumulative Newton and Krylov iteration counts, number of sparse Gaussian factorizations, computational time and memory requirements for the limited memory preconditioner with  $k = 5$  Ritz vectors

Strategy	Standard	Ritz-LMP
Loading steps	9	5
Newton iteration count	42	25
Total Krylov iteration count	2482	1383
Number of sparse Gaussian factorizations	4	1
CPU time	4985	2185
CPU time decrease (%)	×	56
Memory (Mo)	14877	15011
Memory increase (%)	×	0.9

“Standard” corresponds to the solution of the linear system (6) with GMRES(30) combined with the sparse Gaussian factorization based preconditioner  $\mathcal{M}_{sp}$

often, which allows to save one sparse Gaussian factorization through the complete sequence.

Now, in order to minimize the total computational time, we aim at decreasing the total number of loading steps before reaching the complete loading. Therefore, we perform a new simulation with at least 5 increments. The results related to this second experiment are collected in Table 6. We note that the standard solution with only the sparse Gaussian based preconditioner  $\mathcal{M}_{sp}$  requires more than 5 loading steps. This simply means that the Newton’s method does not converge at some steps in the sequence. In such a case, the solution algorithm proposed in *Code\_Aster* automatically defines new closer steps. When the Ritz-LMP strategy is used, only a single sparse Gaussian factorization is used through the

whole sequence, since at each Newton iteration the number of required iterations is kept below the threshold (here 100). This satisfactory behaviour leads to important savings in terms of computational time (56%). This second experiment highlights the efficiency and the robustness of the Ritz-LMP strategy. Finally, we further note that the overcost in terms of memory requirements is again very moderate (at most 0.9%).

## 7 Conclusions

We have proposed a new preconditioner update strategy to be used in combination with a Krylov subspace method for the solution of sequences of large-scale linear systems with nonsingular matrices. This preconditioner update strategy extends earlier formulations [27,28] related to symmetric positive definite and symmetric indefinite matrices, respectively.

The limited memory preconditioner used in combination with approximate spectral information based on Ritz vectors (Ritz-LMP) has proved to be efficient in terms of both preconditioner applications and computational operations on three realistic large-scale problems in structural mechanics, where sequences of saddle point linear systems have to be solved. Numerical experiments have highlighted the relevance of the proposed preconditioner that leads to a significant decrease in terms of computational operations. A saving of up to 56% in terms of computational time—at approximately the same memory cost—is obtained with respect to the original method on one of these large-scale applications. A performance comparison with deflated GMRES has been also proposed on the first two applications. Their performance is found to be quite close, when considering a sequence with a fixed matrix. Nevertheless, in the case of a sequence with varying matrices, the deflation method requires to update projection operators for each new linear system. The limited memory preconditioner (which is fixed through the sequence) has been shown to be more efficient by a factor of 2 in this setting. This shows the relevance of the proposed strategy and we expect this algebraic preconditioner to be efficient in combination with other preconditioners (such as domain decomposition methods) and in other areas of computational science and engineering as well.

To conclude, we mention two possible improvements. First, when solving a sequence of linear systems, recycling subspace information may further improve the convergence rate of the Krylov subspace method for the subsequent linear systems. Hence, it may be worth combining the limited memory preconditioner with such Krylov subspace methods [36,50,61,67]. Secondly, the numerical experiments conducted here have relied on a limited memory preconditioner with columns of  $S$  selected as Ritz vectors associated with the

smallest in modulus Ritz values. This heuristic has proved numerically relevant. Nevertheless, selecting automatically and cheaply how many appropriate column vectors for  $S$  are required to improve the convergence rate of the Krylov subspace method would be certainly a major enhancement. First attempts have been proposed in [21, 23] in the framework of deflation methods for the solution of sequences with symmetric indefinite matrices. We leave both topics for a future study.

**Acknowledgements** The authors thank the two anonymous reviewers for their comments. The authors are indebted to Jed Brown and Barry Smith (PETSc Team) and Thomas de Soza (EDF) for their comments and suggestions related to the implementation of the limited memory preconditioners into PETSc and *Code\_Aster*. The authors would like to acknowledge ANRT (Association Nationale de la Recherche et de la Technologie) for the CIFRE Grant dotation (2012/0687) supporting Sylvain Mercier.

## Appendix

We give in this section the two theorems cited in Sect. 4.1.

**Theorem 1** *Let  $A \in \mathbb{R}^{N \times N}$  be a nonsingular matrix and assume that  $S \in \mathbb{R}^{N \times k}$  is of full column rank  $k$ , with  $k \leq N$  and denote  $\mathcal{S} = \mathcal{R}(S)$ . The preconditioner  $H$  given in Definition 1 is nonsingular if and only if*

$$\mathbb{R}^N = (A\mathcal{S})^\perp \oplus \mathcal{S}. \quad (22)$$

*Proof* Using relation (7), the preconditioner given in Definition 1 can be written as

$$H = P_{(A\mathcal{S})^\perp} + S(S^T A^T A S)^{-1} S^T A^T. \quad (23)$$

Hence, the preconditioned operator  $HA$  simply reads

$$HA = P_{(A\mathcal{S})^\perp} A + I_N - Q. \quad (24)$$

Since  $\mathcal{R}(P_{(A\mathcal{S})^\perp} A) = (A\mathcal{S})^\perp$ ,  $\mathcal{N}(P_{(A\mathcal{S})^\perp} A) = \mathcal{S}$ ,  $\mathcal{N}(I_N - Q) = (A^T A \mathcal{S})^\perp$  and  $\mathcal{R}(I_N - Q) = \mathcal{S}$ , necessary and sufficient conditions for  $HA$  to be invertible are given by

$$\mathbb{R}^N = (A\mathcal{S})^\perp \oplus \mathcal{S} \text{ and } \mathbb{R}^N = (A^T A \mathcal{S})^\perp \oplus \mathcal{S}.$$

We note that the condition  $\mathbb{R}^N = (A^T A \mathcal{S})^\perp \oplus \mathcal{S}$  holds since  $A^T A$  is symmetric positive definite. This completes the proof, since  $A$  is supposed to be nonsingular.  $\square$

**Theorem 2** *Let  $A \in \mathbb{R}^{N \times N}$  be a nonsingular matrix and  $H$  be given by (15) in Definition 1. Assume that the columns of  $W \in \mathbb{R}^{N \times k}$  and  $W_\perp \in \mathbb{R}^{N \times (N-k)}$  form an orthonormal*

*basis for  $A\mathcal{S}$  and  $(A\mathcal{S})^\perp$ , respectively. The spectrum of the preconditioned operator  $HA$  is then given by*

$$\Lambda(HA) = \{1\} \cup \Lambda(W_\perp^T A W_\perp).$$

*Proof* Using relations (7) and (23) leads to

$$HA = AP_{(A\mathcal{S})^\perp} + I_N - P_{(A\mathcal{S})^\perp}. \quad (25)$$

Since the columns of  $[W, W_\perp]$  form an orthonormal basis of  $\mathbb{R}^N$ , we have

$$\Lambda(HA) = \Lambda([W, W_\perp]^T HA [W, W_\perp]) \text{ and}$$

$$[W, W_\perp]^T HA [W, W_\perp] = \begin{pmatrix} W^T A W & W^T A W_\perp \\ W_\perp^T A W & W_\perp^T A W_\perp \end{pmatrix}.$$

Using relation (25) and basic properties of the orthogonal projection  $P_{(A\mathcal{S})^\perp}$ , we finally obtain

$$[W, W_\perp]^T HA [W, W_\perp] = \begin{pmatrix} I_k & W^T A W_\perp \\ 0_{N-k,k} & W_\perp^T A W_\perp \end{pmatrix},$$

which completes the proof.  $\square$

## References

1. Amestoy PR, Duff IS, L'Excellent J-Y (2000) Multifrontal parallel distributed symmetric and unsymmetric solvers. *Comput Methods Appl Mech Eng* 184:501–520
2. Amestoy PR, Duff IS, L'Excellent J-Y, Koster J (2001) A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J Matrix Anal Appl* 23(1):15–41
3. Amestoy PR, Guermouche A, L'Excellent JY, Pralet S (2006) Hybrid scheduling for the parallel solution of linear systems. *Parallel Comput* 32(2):136–156
4. Baglama J, Calvetti D, Golub G, Reichel L (1998) Adaptively preconditioned GMRES algorithms. *SIAM J Sci Comput* 20:243–269
5. Bai Z, Demmel J, Dongarra J, Ruhe A, van der Vorst HA (2000) Templates for the solution of algebraic eigenvalue problems: a practical guide, vol 11. SIAM, Philadelphia
6. Bargellini R (2011) Elasto-visco-plastic modeling with isotropic hardening in large deformations. Technical Report, EDF R&D, AMA R5.03.21
7. Bellavia S, Bertaccini D, Morini B (2011) Nonsymmetric preconditioner updates in Newton–Krylov methods for nonlinear systems. *SIAM J Sci Comput* 33(5):2595–2619
8. Bellavia S, Morini B, Porcelli M (2014) New updates of incomplete LU factorizations and applications to large nonlinear systems. *Optim Methods Softw* 29:321–340
9. Benzi M (2002) Preconditioning techniques for large linear systems: a survey. *J Comput Phys* 182(2):418–477
10. Benzi M, Golub GH, Liesen J (2005) Numerical solution of saddle point problems. *Acta Numer* 14(1):1–137
11. Bonet J, Wood RD (1997) *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press, Cambridge
12. Broyden CG (1969) A new method of solving nonlinear simultaneous equations. *Comput J* 12(1):94–99
13. Carpentieri B, Duff IS, Giraud L (2003) A class of spectral two-level preconditioners. *SIAM J Sci Comput* 25(2):749–765



14. Chapman A, Saad Y (1997) Deflated and augmented Krylov subspace techniques. *Numer Linear Algebra Appl* 4(1):43–66
15. Dostál Z (1988) Conjugate gradient method with preconditioning by projector. *Int J Comput Math* 23(3):315–323
16. Duintjer Tebbens J, Tuma M (2007) Efficient preconditioning of sequences of nonsymmetric linear systems. *SIAM J Sci Comput* 29(5):1918–1941
17. Duintjer Tebbens J, Tuma M (2010) Preconditioner updates for solving sequences of linear systems in matrix-free environment. *Numer Linear Algebra Appl* 17:997–1019
18. Eirola T, Nevanlinna O (1989) Accelerating with rank-one updates. *Linear Algebra Appl* 121:511–520
19. Elman HC, Silvester DJ, Wathen AJ (2014) Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics, 2nd edn. Oxford University Press, Oxford
20. Erlangga Y, Nabben R (2008) Deflation and balancing preconditioners for Krylov subspace methods applied to nonsymmetric matrices. *SIAM J Matrix Anal Appl* 30(2):684–699
21. Gaul A (2014) Recycling Krylov subspace methods for sequences of linear systems : analysis and applications. Ph.D. thesis, TU Berlin, Germany
22. Gaul A, Gutknecht MH, Liesen J, Nabben R (2013) A framework for deflated and augmented Krylov subspace methods. *SIAM J Matrix Anal Appl* 34(2):495–518
23. Gaul A, Schlömer N (2015) Preconditioned recycling Krylov subspace methods for self-adjoint problems. *Electron Trans Numer Anal* 44:522–547
24. Giraud L, Gratton S, Martin E (2007) Incremental spectral preconditioners for sequences of linear systems. *Appl Numer Math* 57:1164–1180
25. Golub GH, Greif C (2003) On solving block-structured indefinite linear systems. *SIAM J Sci Comput* 24(6):2076–2092
26. Gosselet P, Rey C, Pebrel J (2013) Total and selective reuse of Krylov subspaces for the resolution of sequences of nonlinear structural problems. *Int J Numer Methods Eng* 94(1):60–83
27. Gratton S, Mercier S, Tardieu N, Vasseur X (2016) Limited memory preconditioners for symmetric indefinite problems with application to structural mechanics. *Numer Linear Algebra Appl* 23(5):865–887
28. Gratton S, Sartenauer A, Tshimanga J (2011) On a class of limited memory preconditioners for large scale linear systems with multiple right-hand sides. *SIAM J Opt* 21(3):912–935
29. Gutknecht MH (2012) Spectral deflation in Krylov solvers: a theory of coordinate space based methods. *Electron Trans Numer Anal* 39:156–185
30. Gutknecht MH (2014) Deflated and augmented Krylov subspace methods: a framework for deflated BiCG and related solvers. *SIAM J Matrix Anal Appl* 35:1444–1466
31. Ipsen ICF (2001) A note on preconditioning nonsymmetric matrices. *SIAM J Sci Comput* 23(3):10501–1051
32. Jönsthövel TB, Van Gijzen MB, MacLachlan S, Vuik C, Scarpas A (2012) Comparison of the deflated conjugate gradient method and algebraic multigrid for composite materials. *Comput Mech* 50:321–333
33. Jönsthövel TB, Van Gijzen MB, Vuik C, Scarpas A (2013) On the use of rigid body modes in the deflated preconditioned conjugate gradient method. *SIAM J Sci Comput* 35:B207–B225
34. Kelley CT (2003) Solving nonlinear equations with Newton's method, vol 1. SIAM, Philadelphia
35. Kharchenko S, Yerebin A (1995) Eigenvalue translation based preconditioners for the GMRES(k) method. *Numer Linear Algebra Appl* 2:51–77
36. Kilmer M, de Sturler E (2006) Recycling subspace information for diffuse optical tomography. *SIAM J Sci Comput* 27(6):2140–2166
37. Liesen J, Strakoš Z (2013) Krylov subspace methods principles and analysis. Oxford University Press, Oxford
38. Mercier S (2015) Fast nonlinear solvers in solid mechanics. Ph.D. thesis, University of Toulouse, Toulouse, France, <http://thesesups-ups-tlse.fr/3000/1/2015TOU30305.pdf>
39. Meyer CD (2000) Matrix analysis and applied linear algebra. SIAM, Philadelphia
40. Michel-Ponnelle S (2013) Modeling of the pretension cables. Technical report, EDF R&D, AMA R7.01.02
41. Morales JL, Nocedal J (2000) Automatic preconditioning by limited memory quasi-Newton updating. *SIAM J Opt* 10(4):1079–1096
42. Nash S (1984) Newton-type minimization via the Lanczos method. *SIAM J Numer Anal* 21:770–778
43. Nečas J, Hlaváček I (1980) Mathematical theory of elastic and elastoplastic bodies. Elsevier, Amsterdam
44. Nicolaides RA (1987) Deflation of conjugate gradients with applications to boundary value problems. *SIAM J Numer Anal* 24(2):355–365
45. Nocedal J, Wright SJ (1999) Numerical optimization. Springer series in operations research and financial engineering. Springer, Berlin
46. Notay Y (2014) A new analysis of block preconditioners for saddle point problems. *SIAM J Matrix Anal Appl* 35(1):143–173
47. O'Leary DP, Yerebin A (1994) The linear algebra of block quasi-Newton algorithms. *Linear Algebra Appl* 212(213):153–168
48. Olshanskii MA, Tyrtyshnikov EE (2014) Iterative methods for linear systems: theory and applications. SIAM, Philadelphia
49. Paige CC, Parlett BN, van der Vorst HA (1995) Approximate solutions and eigenvalue bounds from Krylov subspaces. *Numer Linear Algebra Appl* 2:115–134
50. Parks M, de Sturler E, Mackey G, Johnson D, Maiti S (2006) Recycling Krylov subspaces for sequences of linear systems. *SIAM J Sci Comput* 28(5):1651–1674
51. Pestana J, Wathen AJ (2015) Natural preconditioning and iterative methods for saddle point systems. *SIAM Rev* 57(1):71–91
52. Proix JM (2013) Elasto-visco-plastic modeling of Chaboche-type. Technical Report, EDF R&D, AMA R5.03.04
53. Rey C, Risler F (1998) A Rayleigh–Ritz preconditioner for the iterative solution to large scale nonlinear problems. *Numer Algorithms* 17:279–311
54. Risler F, Rey C (1998) On the reuse of Ritz vectors for the solution to nonlinear elasticity problems by domain decomposition methods. *Contemp Math* 218:334–340
55. Saad Y (2003) Iterative methods for sparse linear systems. SIAM, Philadelphia
56. Saad Y, Schultz MH (1986) GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J Sci Stat Comput* 7(3):856–869
57. Saad Y, Yeung M, Erhel J, Guyomarc'h F (2000) A deflated version of the conjugate gradient algorithm. *SIAM J Sci Comput* 21(5):1909–1926
58. Schnabel RB (1983) Quasi-Newton methods using multiple secant equations. Technical report, Technical report CU-CS-247-83, Department of Computer Sciences, University of Colorado, Boulder, CO
59. Shewchuk J (2002) What is a good linear finite element ? Interpolation, conditioning, anisotropy, and quality measures, University of California at Berkeley, unpublished preprint
60. Simoncini V, Szyld DB (2007) Recent computational developments in Krylov subspace methods for linear systems. *Numer Linear Algebra Appl* 14(1):1–59
61. Soodhalter KM, Szyld DB, Xue F (2014) Krylov subspace recycling for sequences of shifted linear systems. *Appl Numer Math* 81:105–118
62. Tang JM, MacLachlan SP, Nabben R, Vuik C (2010) A comparison of two-level preconditioners based on multigrid and deflation. *SIAM J Matrix Anal Appl* 31(4):1715–1739

63. Tardieu N, Cheignon E (2012) A Newton–Krylov method for solid mechanics. *Eur J Comput Mech* 21(3–6):374–384
64. Toselli A, Widlund O (2005) Domain decomposition methods: algorithms and theory. Springer, Berlin
65. van der Vorst HA (2003) Iterative Krylov methods for large linear systems. Cambridge University Press, Cambridge
66. Vassilevski PS (2008) Multilevel block factorization preconditioners: matrix-based analysis and algorithms for solving finite element equations. Springer, New-York
67. Wang S, de Sturler E, Paulino G (2007) Large-scale topology optimization using preconditioned Krylov subspace methods with recycling. *Int J Numer Methods Eng* 69(12):2441–2468
68. Wathen AJ (2015) Preconditioning. *Acta Numer* 24:329–376
69. Yang UM (1995) A family of preconditioned iterative solvers for sparse linear systems. Ph.D. thesis, University of Illinois at Urbana-Champaign
70. Yang UM, Gallivan KA (1995) A new family of preconditioned iterative solvers for nonsymmetric linear systems. *Appl Num Math* 19(3):287–317